

F-013

## ABLAにおけるノード情報伝達に基づくモバイルエージェントの移動先決定方式 A Movement Decision Method for Mobile Agents Based on Conveyed Message of Node on ABLA

野地 脩宏†  
Naohiro Noji

加藤 貴司†  
Takahi Katoh

ベッド B. ビスタ†  
Bhed Bahadur Bista

高田 豊雄†  
Toyoo Takata

### 1. はじめに

近年、インターネット上の脅威活動が爆発的な増加傾向を見せており、ネットワーク管理者等は、常に最新の攻撃動向やインターネットのトラフィック状況等を把握し、マルウェアや攻撃元の早期発見を行うことで、攻撃・被害拡大の防止に努めなければならない[1]。その一つの手段として、インターネット上で発生している攻撃の動向やトラフィックの発生状況の観測を目的とした、分散型インターネット観測システム ABLA (Agent Based Log Analyzing System) が提案・開発されている[2]。

この ABLA において、観測情報収集時のエージェント移動先決定の際、2 ホップ以上先のエージェント要求を満たす観測点を考慮できない、エージェント帰還の際に、経由観測点が消失すると帰還困難になる、という問題がある。そこで本稿では、これらの問題を解決するために、各ノードのノード情報と、各ノード間のリンク及びそれらの経路情報を、エージェントによって各ノードへ伝達する手法を提案し、これを利用したエージェントの移動先決定方式を提案する。

## 2. ABLA (Agent Based Log Analyzing System)

### 2.1 ABLA 概要

ABLA は Peer-to-Peer (以下 P2P) ネットワークとモバイルエージェント (以下エージェント) を用い、参加する全ての計算機を観測点 (以下ノード) とする分散型のインターネット観測システムである。

利用者は観測情報を収集する際に、自分の要求に合わせたエージェントを発行する。発行されたエージェントはノードを移動し、移動先の各ノード上でログ情報を解析・結果収集を行う (図 1)。このため、利用者は自分の要求に合った観測情報を収集することができ、多数の観測点のログ情報を 1 ヲ所に集積することなく解析を行うことが可能である。

### 2.2 観測点の匿名化

ABLA では各ノード (ログ情報提供者) の匿名化のために、IP アドレスの代わりに固有のハッシュ値を ID として利用している。IP アドレスを使用しているのは、直接接続を確立している他ノードとの通信のみであり、これが隣接ノードとなっている。

エージェントの移動は、ABLA 内でエージェントの転送機能を提供するコンポーネントが、直列化したエージェントを移動先ノードに通信するため、エージェント自身は IP アドレスを使用することはない。このことによって、エージェントに対して IP アドレスの隠ぺいを実現している。

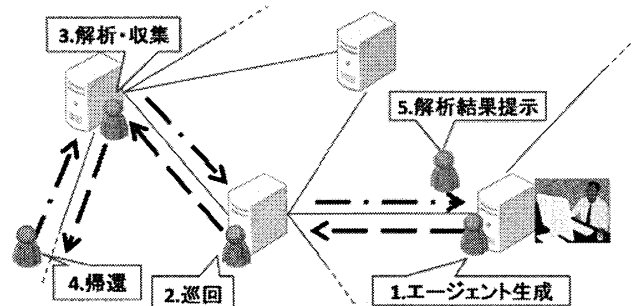


図 1: ABLA 概要

### 2.3 観測情報収集時のエージェント移動先ノードの決定方法

ABLA では観測情報収集時に利用者がエージェントを発行し、エージェントが代行して収集を行う。エージェントは移動先を決定する際、各隣接ノードのノード情報を参照する。このノード情報から最も自身の要求条件に合った隣接ノードを選択し、移動する (図 2)。

ノード情報とは、ノード自身の概要と、所持している観測情報の概要である。ノード情報は他ノードとの接続確立時に取得し、定期的に更新する。

#### ノード情報

- ノード ID
- 観測情報の作成日時
- 観測情報の更新日時
- 観測情報の種類 (リソースタイプ)
- 接続ノード数
- 観測情報の量 (ログファイルサイズ)

#### ノード情報によるエージェントの移動先決定方法

エージェントは以下の条件を満たす隣接ノードを移動先として選択する。

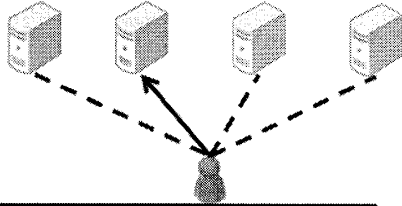
1. 観測情報の作成日時・更新日時が、エージェントの指定時間帯を満たしていること。
2. リソースタイプがエージェントの要求と合致していること。
3. まだ訪れていないノードであること。

エージェントの要求条件を満たすノードが存在しない、もしくは複数存在した場合は以下のように処理を行う。

- 条件を満たすノードが存在しなかった場合
  - ▶ 最も接続ノード数の多い隣接ノードへ移動し、再度移動先の決定を行う。
- 条件を満たすノードが複数存在した場合
  - ▶ 最もログファイルサイズが大きいノードを移動先として決定する。
  - ▶ エージェントは実際にノードで解析を行うまで、ログの内容 (宛先ポートや送信元ホスト等、ノード情報以外の情報) を知ることが出来ない。このことか

†岩手県立大学大学院ソフトウェア情報学研究科

提示情報	観測点A	観測点B	観測点C	観測点D
ログ作成日時	2010/2/1	2010/2/6	2010/2/9	2010/1/28
ログ更新日時	2010/2/8	2010/2/12	2010/2/12	2010/2/12
リソース	TCPDUMP	TCPDUMP	TCPDUMP	TCPDUMP
接続ホスト数	5	3	4	7
ログファイルサイズ	1300KB	800KB	500KB	2000KB



判断要素	エージェント要求値
指定時間帯	2010/2/10~2010/2/12
リソースタイプ	TCPDUMP
解析済み観測点	観測点D, G, H, I

図2: エージェントの移動先決定方法

ら、利用者が要求を満たす情報を得る可能性を高くするために、よりログファイルサイズの大きいノードを選択する。

#### 2.4 観測情報収集後のエージェント帰還方法

エージェントは観測情報収集後に、以下の手順でエージェント発行元に帰還する。

1. エージェントは観測情報の解析・収集時に、経由したノードのIDをリストとして保持しておく。
2. 保持したIDリストを参照し、逆順にIDの合致したノードを辿り、帰還する。ただし、IDリスト内で経路の短縮が出来る場合は、それを利用する。

2.2節で述べた通り、エージェントは移動の際、IPアドレスを使用しない。従って、帰還時においてもエージェントはIPアドレスを使用した帰還移動はできず、発行元のIPアドレスを利用して直接帰還することは出来ない(図3)。

#### 2.5 ABLAにおけるエージェントの移動先決定の問題点

##### 2.5.1 エージェントの移動先ノードの決定

2.3節で述べた通り、従来のABLAにおけるエージェントは、隣接ノード情報のみを参照し、移動先を決定しているため、2ホップ以上先のノードを全く考慮出来ない。そのため、2ホップ以上先にエージェントにとって更に要求を満たす情報を持つノードが存在したとしても、それを見逃して別経路に移動してしまう可能性がある。

##### 2.5.2 エージェントの帰還

ABLAは個人ユーザが観測点として参加していることから、離脱や障害等によるノードの消失が頻繁に発生することが考えられる。しかし2.4節で述べた通り、帰還時のエージェントは、経由したノードを逆順に辿ることで帰還を行っていることから、もし帰還時にIDリスト内のノードが消失してしまった場合、帰還することは不可能となる。このときIDリスト以外に帰還可能な経路があったとしても、その経路を知ることが出来ず、帰還不可能な状況に陥ってしまう可能性がある。

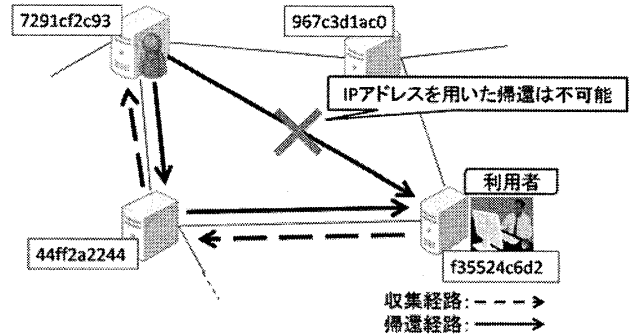


図3: エージェントの帰還方法

### 3. エージェントによる提示情報伝達に基づく移動先決定方式の提案

2.5節で述べた問題点に対し、2ホップ以上先のノードを考慮した移動先決定や帰還が可能な、エージェントの移動を実現する。しかしそのためには、各ノードがネットワークポロジを知らなければならない。

そこで、本稿ではまず、ノード情報とリンク情報を提示情報として各ノードへ伝達する手法を提案する。次に、これを利用してエージェントの移動先決定方式を提案する。更に、本提案手法の問題点とその解決手法について述べる。

#### 3.1 提示情報の伝達と保持

##### 3.1.1 各エージェントによる提示情報の伝達

各エージェントはノード巡回の際に、以下のような手順で提示情報の伝達を行う。尚、各ノードはNTP等によって時刻を同期しており、時刻情報は全て正確であるとする。

1. エージェントは、滞在しているノードから提示情報を取得する。
2. エージェントは自身が所持している提示情報を更新する。
3. エージェントは次の移動先ノードへ移動する。
4. 移動後、エージェントは所持している提示情報を、移動先ノードへ伝える。

#### 提示情報

エージェントが各ノードから取得する提示情報は以下の項目である。

- ノード情報
  - 現在滞在しているノード、及びその隣接ノードの情報
    - ▶ ノードID
    - ▶ 観測情報作成日時
    - ▶ 観測情報更新日時
    - ▶ リソースタイプ
    - ▶ 接続ノード数
    - ▶ ログファイルサイズ
    - ▶ ノード情報取得時のタイムスタンプ。
- リンクリスト
  - ▶ リンクリスト情報
    - ◇ 現在滞在しているノードと、その隣接ノードのリンク情報。
  - ▶ リンクリスト日時
    - ◇ エージェントによるリンクリスト取得・更新時のタイムスタンプ。

**提示情報の更新**

エージェントは提示情報取得時、以下の手順で所持している提示情報の更新を行う。

1. ノード情報、リンクリスト取得時、同じノードの情報を、エージェントが所持していない場合、新規のノード情報、リンクリストとして、それぞれの項目に追加する。
2. ノード情報取得時、既に同じノードのノード情報を、エージェントが所持していた場合、取得日時の古いノード情報を、新しいノード情報で上書きする。
3. リンクリスト取得時、既に同じノードのリンクリストを、エージェントが所持していた場合、取得日時の古いリンクリストを、新しいリンクリストで上書きする。

**エージェントによる提示情報伝達の例 (図4)**

出発点をSとしたエージェントが、A, B, C, Dと移動した時、AにはSとその隣接ノードのノード情報、及びSのリンクリストを提示する。DではS, A, B, Cとそれらの隣接ノードのノード情報、及びS, A, B, Cのリンクリストを提示する。

**3.1.2 各ノードにおける提示情報の保持**

各ノードでは、エージェントから伝達された提示情報を、以下のように扱う。

1. ノードが保持している提示情報を更新する。
2. 更新したリンクリストを基に、IDによる経路情報を作成・更新する。

**提示情報の更新**

伝達された提示情報から、ノードは以下の手順で所持している提示情報の更新を行う。

1. ノード情報、リンクリスト取得時、同じノードの情報を所持していない場合、新規のノード情報、リンクリストとして、それぞれの項目に追加する。
2. ノード情報取得時、既に同じノードのノード情報を所持していた場合、取得日時が新しいノード情報を使用するため、古いノード情報は削除する。
3. リンクリスト取得時、既に同じノードのリンクリストを所持していた場合、取得日時が新しいリンクリストを使用するため、古いリンクリストは削除する。
4. リンクリスト取得時、一方では隣接ノードとして認識しているにも関わらず、その隣接ノードでは、先のノードを隣接ノードと認識していない、等のように、リンクリスト間の不整合が発生した場合、以下のように更新を行う。
  - リンクリストの日時から、新しいリンクリストの情報に沿って、古いリンクリストで隣接ノードの追加・削除を行う。
    - ▶ 例：[A : B, C, D] (新) , [B : E, F] (古) , ならば、新たにB-Aのリンクが張られたと分かるので、リンクリストBを[B : A, E, F]とする。
    - ▶ 例：[A : B, C, D] (古) , [B : E, F] (新) , ならば、B-Aのリンクが切れたと分かるので、リンクリストAを[A : C, D]とする。
5. リンクリストの更新によって、隣接ノードとの対応が無くなったリンクリスト (自身のIDしかないリンク

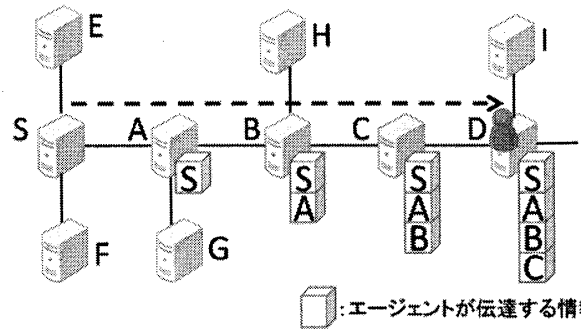


図4: エージェントによる提示情報伝達の例

リスト)が発生した場合、そのノードは消失したと見なし、そのノード情報とリンクリストを削除する。

**各ノードにおける経路情報の作成**

取得したリンクリストを用い、ノードは自身を中心とした経路情報 (各ノードまでの最短経路とそのホップ数) を作成する。

尚、最短経路が複数ある場合、経由する各ノード情報の、総接続ノード数が多い経路を選択する。これは、途中経路に障害があった時、複数の経路を選択可能とするためである。更に、総接続ノード数が同数であった場合は、経路候補の中から、ランダムにひとつの経路を選択する。

**各ノードにおける提示情報保持の例 (図5, 表1)**

Hに着目した場合、Aから出発し、C, E, Hと辿ったエージェントから、提示情報をHが受け取ると、HはA, C, Eのノード情報とそのリンクリスト、及びそれらの隣接ノードB, D, Fのノード情報を把握する。また、その情報によって表1の経路情報が作成される。

**3.2 提示情報伝達に基づくエージェントの移動方式**

提示情報の伝達を行うことによって、各ノードは隣接ノード以外の他ノード情報、及びそのリンクを知ることが可能となる。このことにより、エージェントは移動先を決定する際に、ノードからそれらの情報を参照することで、隣接ノード及びIDリスト以外のノードを考慮し、移動することが可能となる。

ここで、エージェントが滞在ノードの持つ提示情報を参照し、隣接ノード以外のノードを考慮した移動方式、及びそれによって得られる効果について述べる。

**3.2.1 2ホップ以上先のノードを考慮した解析対象ノードの決定**

2ホップ以上先のノードを解析対象として決定し、移動するまでの手順を以下に述べる。

1. エージェントは現在滞在しているノードの持つ提示情報から、他ノードを参照する。
2. 1.で受け取ったノード情報を基に、2.3節のエージェントの移動先決定方法により、解析対象を決定する。
3. 解析対象までのホップ数によって、エージェントは以下のように移動を行う。
  - 解析対象が隣接ノードである場合
    - ▶ 既存のシステムと同様に隣接ノードへ移動を行う。
  - 解析対象が2ホップ以上先のノードである場合

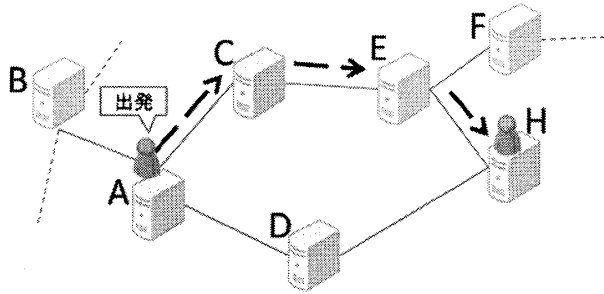


図5: 各ノードにおける提示情報保持

表1: 観測点Hが持つ経路情報

ノードID	隣接ID	最短経路	ホップ数
H	E, D	H	0
E	C, F, H	H:E	1
D	A, H	H:D	1
F	E	H:E:F	2
C	A, E	H:E:C	2
A	B, C, D	H:D:A	2
B	A	H:D:A:B	3

A) エージェントは、現在滞在しているノードから、解析対象となるノードまでの最短経路を受け取り、解析対象経路リスト（帰還時に使用する ID リストとは別のリスト）として所持する。

B) エージェントは、そのリストに従い、解析対象ノードまで移動する。  
現在居るノードの ID が、解析対象経路リストの最終 ID でなければ、経由ノードであると判断し、経由ノードでは、エージェントの観測情報の解析・収集は行わない。（提案手法による提示情報の伝達は行う。）

C) エージェントは、1 ホップ移動する度に、滞在しているノードから、解析対象までの最短経路を取得し、現在所持している解析対象経路リストよりホップ数が短ければ、新しく取得した最短経路を解析対象経路リストとする。

4. エージェントは解析対象のノードに到着後、観測情報の解析・収集を行う。

解析対象までの移動途中、エージェントの要求によりふさわしいノードを発見したとしても、解析対象は変更しない。理由は以下の通りである。

- 発見したノードが 2 ホップ以上先であり、移動中に辿り着けなくなる等、エージェントが移動だけを繰り返すことを防ぐため。
- エージェントによる提示情報の伝達によって、解析・結果収集後、次の解析対象を選択する際に、エージェントはそのノードを解析対象候補とすることが可能なため。

以上の手順によって、エージェントは隣接ノード以外のノードを解析対象として決定することが可能となり、より自身の要求条件と合致したノードを解析対象としながら巡回することが出来る。

### 3.2.2 各ノードが持つ経路情報による帰還経路の決定

各ノードは、各エージェントから伝達された情報を基に、経路情報を作成する。ノードは、この経路情報から複数の経路を把握することが可能である。帰還時のエージェントは、以下のように帰還経路を選択し、帰還移動を行う。

1. エージェントは、現在滞在しているノードから、自身の発信元までの最短経路を受け取り、帰還 ID リストとして所持する。
2. 帰還 ID リストに沿って帰還を行う。
3. 1 ホップ移動する度に、現在エージェントが持つ帰還 ID リストと、現在滞在しているノードの経路情報内のエージェント発信元までの最短経路を比較し、より最短となる経路があれば、それを帰還 ID リストとして選択する。

以上の手順によって、既存システムの ID リストによる帰還経路のみではなく、複数の帰還経路を確保することが可能となり、一つの経路が帰還不可能となった場合でも、他の経路を帰還経路として選択することが出来る。また、複数の経路を把握出来ることから、最短経路を選択することで、帰還経路の短縮が可能となる。

### 3.3 提示情報伝達による移動の問題とその対策

提示情報の伝達によって各ノードが得られる情報は、他エージェントが辿ってきた時点の情報であり、隣接ノード以外の他ノードの情報は過去の情報である。そのため、エージェントが 2 ホップ以上先のノードを移動先として定めた時、そのノードが既に消失している可能性がある。

この問題に対して、エージェントは次のような行動を取る。

#### 3.3.1 観測情報収集時のエージェント

- 解析対象としたノードが消失している場合
  - A) 消失発見時点のノードにおいて、エージェントは解析対象の再決定を行う。（決定方法は 3.2.1 節の通りである。）
- 解析対象経路リストのリンクが既に切れており、解析対象まで辿りつけない場合
  - A) エージェントは、滞在ノードの経路情報から、解析対象までの別経路があれば教えてもらい、それを解析対象経路リストにする。
  - B) 滞在ノードの経路情報から別経路が得られない場合、滞在ノードで解析対象の再決定を行う。

#### 3.3.2 帰還時のエージェント

移動中のノードにて、帰還 ID リスト通りのリンクが発見できない場合、エージェントは、移動後のノードの経路情報から、帰還経路を再探索し、帰還 ID リストの変更を行う。

## 4. 提示情報伝達による移動方式を適用したエージェント移動の有効性評価

### 4.1 実験

本提案手法の有効性を評価するために、以下の実験を行った。

1. エージェントの解析対象となるノードのログファイルサイズ総計

ここで、ログファイルサイズの総計によって評価するのは、エージェントが移動先を決定する際、ログファイルサイズが大きいノードを選択出来ることが望ましいからである (2.3 節)。

## 2. エージェント帰還成功率

### 実験環境

本実験はシミュレータにより行った。  
シミュレーションパラメータは表 2 の通りである。

### 実験における初期動作

シミュレータにおいて初期参加ノード配置時、全てのノードは隣接ノードしか把握しておらず、この時エージェントを発行した場合、エージェントは隣接ノードと辿ったノードのみを参照するため、本提案手法の効果を十分に発揮出来ず、既存の ABLA システムと本提案手法を適用した ABLA システムとの明確な比較結果を得ることは難しい。

今回の実験では、既存の ABLA システムと本提案手法を適用した ABLA システムとの比較結果をより明確にするため、初期参加ノードの 50% に 1 度エージェントを発行させ、収集・帰還を行わせる (解析を行うノード数は表 4.2 に準ずる)。このことにより、少なくとも 50% のノードは隣接ノード以外の他ノードの提示情報を保持している状況を設定する。

尚、初期エージェント発行ノード数を 50% とした理由は、ABLA では観測情報の収集を目的とする利用者が過半数であると想定しており、50% のノードでのみ観測情報収集が行われなかった場合においても、本提案手法の効果が十分に得られることを確認するためである。

## 4.2 実験 1: エージェントの解析対象となるノードの

### ログファイルサイズ総計

エージェントは移動を行う際に、移動先のノードがエージェントの解析対象条件を複数満たしている場合、移動先候補内で最もログファイルサイズの大きいノードを、移動先として決定している (2.3 節)。このことから、本提案手法を適用することによって、ログファイルサイズの大きいノードを巡回可能であることを示すため、エージェントが解析を行うノードのログファイルサイズの総計を求め、既存の ABLA システムとの結果比較を行う。

### 4.2.1 実験手順

- 3.3 節で述べた本提案手法の問題に対して対策されていることを示すため、ノード消失率に準じてランダムにノードを消失させる。
- ランダムに一つのノードからエージェントを発行する。
- エージェント帰還後、エージェントの解析対象となったノードのログファイルサイズの総計を求め、結果を比較する。

### 4.2.2 実験結果

本提案手法を適用したシステムによるログファイルサイズ総計の向上率 (図 6)

結果から、本提案手法を適用したシステムでは、ノード消失率が 10%~50% の場合に、ログファイルサイズの総計が向上した結果が得られた。更に、ノード消失率が上昇した場合においても向上した結果が得られたことから、3.3 節で提起した問題を解決し、自身の要求条件に合致したノードを再探索出来たとと言える。

表 2: シミュレーションパラメータ

参加ノード数	200
ログファイルサイズ(KB)	100-2000 (ランダム)
各ノード初期リンク数	3
ログ解析ノード数	5

### 本提案手法を適用したシステムによる総ホップ数 (図 7)

各消失率のホップ数と、総計結果を対応させると、ホップ数が多くても総計向上率が低い結果や、少ないホップ数でも総計向上率が高い結果があり、ホップ数が多くなるとログファイルサイズの総計も大きくなるというような、必然的な関係は見られなかった。特に消失率 0% と 10% に着目した場合、消失率 0% よりも 10% の方が平均向上率が上昇し、平均総ホップ数が減少している。

本提案手法を適用したシステムでは、ノード消失率 10%~50% のいずれの場合にも、解析ホップ数より帰還ホップ数の方が少なく、帰還経路の短縮に成功しており、総ホップ数の増加を抑えている。

既存の ABLA システムの平均総ホップ数が 9 であることから (図 8)、本提案手法を適用したシステムによる平均総ホップ数では、2 倍以下に抑えられている。これは本提案手法の帰還経路短縮によって、帰還ホップ数が削減され、総ホップ数の無駄な増加を抑えることが出来たためだと言える。

### 4.2.3 評価

エージェントによる提示情報の伝達により、ノードが隣接ノード以外のノードを把握し、エージェントがその情報を参照することで、エージェントは 2 ホップ以上先のノードを解析対象として選択し、より観測情報の多いノードを巡回することが可能となった。これによりエージェントの移動において、利用者要求を満たすノードの見逃しが改善されていることを示した。よって本提案手法は、観測情報収集時のエージェントの解析対象選択において有効であると言える。

## 4.3 実験 2: エージェント帰還成功率

本提案手法を適用することによって、エージェントが帰還を行う際に、ID リスト内のノード消失が発生した場合においても、別経路による帰還が可能であることを示すため、エージェントの帰還成功率を測定し、既存の ABLA システムでの結果と比較する。

### 4.3.1 実験手順

- ランダムに一つのノードからエージェントを発行する。
- 帰還移動開始時に、ノード消失率に準じてランダムにノードを消失させる。
- エージェントの帰還を確認し、帰還成功率を比較する。
  - 帰還成功率は、100 回のエージェントの帰還移動のうち、帰還に成功した回数の割合である。

### 4.3.2 実験結果

本提案手法を適用したシステムによる帰還成功率の向上率 (表 3, 図 9)

結果から、本提案手法を適用したシステムでは、ノード消失率が 10%~50% の場合に帰還成功率が向上していることが分かる。これは本提案手法によって、移動したノードにおいて帰還経路を選択することが可能なことから、ID リスト以外の別経路によって帰還が行えたことによる結果だと言

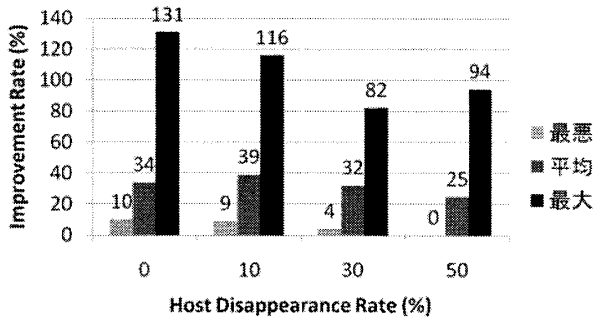


図 6: ログファイルサイズの総計向上率

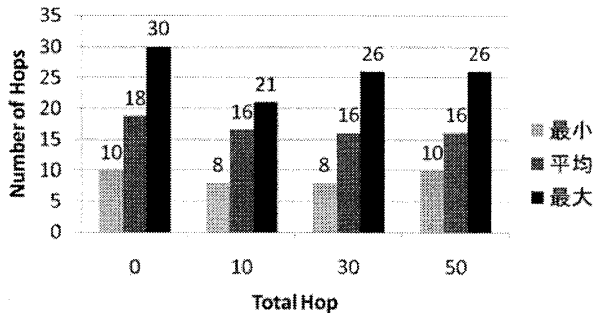


図 7: 本提案手法における総ホップ数

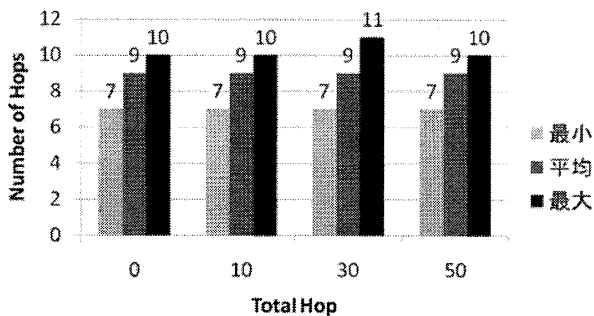


図 8: 従来手法における総ホップ数

表 3: 消失率毎の平均帰還成功率

	消失率 10%	消失率 30%	消失率 50%
従来手法	79.4%	42.2%	4.3%
本提案手法	92.6%	77.1%	11.3%

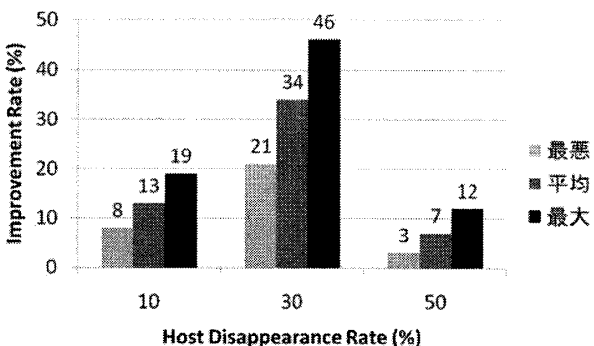


図 9: 帰還成功率の向上率

える。特にノード消失率 30%時に大きく向上していることから、ノード消失率 30%時が、本提案手法の最も有効な状況であると言える。しかしながら、ノード消失率 50%においても帰還成功率が向上していることから、本提案手法による効果は、エージェントの帰還において有効であると言える。

### 4.3.3 評価

エージェントによる提示情報の伝達により、ノードが 2 ホップ以上先のノードまでの経路を把握することで、エージェントがそれを参照することにより、経由ノード消失時においても別経路による帰還が可能であることを示した。よって本提案手法は、ABLA のエージェント帰還成功率向上に有効であると言える。

## 5. おわりに

本稿では、ABLA のエージェント移動において、2 ホップ以上先に更に要求を満たす情報を持つノードが存在しても、それを見逃して別経路を選択してしまう可能性や、ID リストによる帰還が不可能となり、別経路による帰還が出来ない可能性があるという問題を、エージェントが行う提示情報の伝達によって改善する手法を提案した。本提案手法によって、ノードが 2 ホップ以上先のノードを把握出来るようになり、エージェントがその情報を参照することで、観測情報収集時のエージェント要求条件に合致した、より適切な解析対象となるノードの決定や、ID リスト以外の別経路選択による帰還成功率の向上を実現し、その有効性をシミュレーションによる実験で示した。

今後の課題として、本提案手法を適用した場合、エージェントやノードは、解析結果の他に、提示情報を保持することになり、それに伴うトラフィックや情報量の増加が、ABLA ネットワークに与える負荷を評価し、明確にすることがあげられる。

## 謝辞

本研究は科研費(若手(B) 21700084)の助成を受けたものである。

## 参考文献

- [1] S. Singh, C. Estan, G. Varghese, S. Savage: *The Early Bird System for Real-time Detection of Unknown Worms*: UCSD Tech. Report, CS2003-0761, 2003.
- [2] 葛野弘樹, 中井優志, 渡辺集, 川原卓也, 加藤貴司, Bhed Bahadur Bista, 高田豊雄: モバイルエージェントを用いた分散型インターネット観測システムの提案: 情報処理学会論文誌, Vol.47, No.5, pp.1393-1405 (2006).